Types

Let $x$ range over a set of identifiers and $\alpha$ range over a set of type variables. Now suppose we have a set of types, $\tau$, and a set of type schemes, $\sigma$, given by

$$
\tau ::= \alpha \mid \tau \rightarrow \tau \mid \tau\text{ list}
$$

$$
\sigma ::= \forall \alpha_1, \ldots, \alpha_n (\tau)
$$

and a language of terms, $M$, given by

$$
M ::= x \mid \lambda x (M) \mid MM \mid \text{let } x = M \text{ in } M \mid \text{nil} \mid M :: M
$$

$$
\mid \text{case } M \text{ of } \text{nil} \Rightarrow M \mid x :: x \Rightarrow M
$$

(a) Define the relation of specialisation, $\tau \prec \sigma$, between types $\tau$ and type schemes $\sigma$. [3 marks]

(b) Give the ML-like type inference rules for judgements of the form $\Gamma \vdash M : \tau$, and explain why $\lambda$-bound variables cannot be used polymorphically within a function abstraction, while $\text{let}$-bound variables can within a local declaration.

Hint: Consider the terms $\lambda f (f f)$ and $\text{let } f = \lambda x (x) \text{ in } f f$. [8 marks]

(c) Briefly explain what is meant by capture-avoiding substitution for type schemes. [2 marks]

(d) Prove that for all $\tau$, all $\sigma$ and all substitutions for type schemes $S$, if $\tau \prec \sigma$ holds, then also $S(\tau) \prec S(\sigma)$.

Hint: Use the following property of simultaneous substitution:

$$
(\tau[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]) [\vec{\tau}'/\vec{\alpha}'] = \tau[\vec{\tau}'/\vec{\alpha}'][\tau_1[\vec{\tau}'/\vec{\alpha}']/\alpha_1, \ldots, \tau_n[\vec{\tau}'/\vec{\alpha}']/\alpha_n]
$$

which holds, provided that for each $i$, $\alpha_i$ is distinct from the type variables $\vec{\alpha}'$, and $\alpha_i$ does not occur in type schemes $\vec{\tau}'$. [7 marks]